A set of symbols supported by EB2C & EB2C++

April 1, 2011

Event-B	'C' & 'C++' Language	Comment
n..m	int	Interger type
x ∈ Y	Y x;	Scaler declaration
x ∈ tl_int16	int x;	'C' & 'C++' Context declaration
x ∈ n..m → Y	Y x [m+1];	Array declaration
x :∈ Y	/* No Action */	Indeterminate initialization
x : | Y	/* No Action */	Indeterminate initialization
x = y	if(x==y) {	Conditional
x ≠ y	if(x!=y) {	Conditional
x < y	if(x<y) {	Conditional
x ≤ y	if(x<=y) {	Conditional
x > y	if(x>y) {	Conditional
x ≥ y	if(x>=y) {	Conditional
(x>y) ∧ (x≥z)	if ((x>y) && (x>=z) {	Conditional
(x>y) ∨ (x≥z)	if ((x>y) || (x>=z) {	Conditional
x := y + z	x = y + z;	Arithmetic assignment
x := y - z	x = y - z;	Arithmetic assignment
x := y * z	x = y * z;	Arithmetic assignment
x := y ÷ z	x = y / z;	Arithmetic assignment
x := F(y)	x = F(y);	Function assignment
a := F(x↦y)	a = F(x, y);	Function assignment
x := a(y)	x = a(y);	Array assignment
x := y	x = y;	Scalar action
a := a ⩤ {x↦y}	a(x) = y;	Array action
a := a ⩤ {x↦y} ⩤ {i↦j}	a(x)=y; a(i)=j;	Array action
X⇒Y	if(!X || Y){	Logical Implication
X⇔Y	if((!X || Y) && (!Y || X)){	Logical Equivalence
¬x<y	if(!(x<y)){	Logical not
x ∈ ℕ	unsigned long int x	Natural numbers
x ∈ ℤ	signed long int x	Integer numbers
∀	/* No Action */	Quantifier
∃	/* No Action */	Quantifier
Sets	Supported by C++	Using STL library based Sets operations
Set1	set <*data type*> Set1	STL library
∪	set_union(...)	STL library
∩	set_intersection(...)	STL library
−	set_difference(...)	STL library
⊂	if (includes(...)){	STL library
⊆	if (includes(...) || equal (...)){	STL library
⊄	if (!(includes(...))){	STL library
⊈	if (!(includes(...) || equal (...))){	STL library
fun ∈ ℕ × ℕ → ℕ	long int fun(unsigned long int $arg1$, unsigned long int $arg2$) { //TODO: Add your Code return; }	Function Definition

Table 1: Event-B to C & C++ translation syntax